Modificatory Provisions Detection: a Hybrid NLP Approach

Modificatory Provisions Detection: a Hybrid NLP Approach

Davide Gianfelice
Dipartimento di Informatica
Università di Torino
Corso Svizzera 185,
10149 - Torino, Italy
davide1184@gmail.com

Leonardo Lesmo
Dipartimento di Informatica
Università di Torino
Corso Svizzera 185,
10149 - Torino, Italy
lesmo@di.unito.it

Monica Palmirani
CIRSFID
Università di Bologna
Via Galliera, 3
40121 Bologna, Italy
monica.palmirani@unibo.it

Daniele Perlo
Dipartimento di Informatica
Università di Torino
Corso Svizzera 185,
10149 - Torino, Italy
perlo.daniele@gmail.com

Daniele P. Radicioni*
Dipartimento di Informatica
Università di Torino
Corso Svizzera 185,
10149 - Torino, Italy
radicion@di.unito.it

ABSTRACT

In the last few years University of Turin and CIRSFID University of Bologna collaborated to pair NLP techniques and legal knowledge to detect modificatory provisions in normative texts. Annotating these modifications is a relevant and interesting problem, in that modifications affect the whole normative system; and legal language, though more regular than unrestricted language, is sometimes particularly convoluted, and poses specific linguistic issues. This paper focuses on two major aspects. First, we explore a combination between parsing and regular expressions; to the best of our knowledge, such hybrid strategy has never been proposed before to tackle the problem at hand. Secondly, we significantly extend past works coverage (basically focussed on *substitution*, *integration* and *repeal* modifications) in order to account for further twelve modification kinds. For the sake of conciseness, we fully illustrate and discuss only few modification types that are more relevant and interesting: *suspension*, *prorogation of efficacy*, *postponement of efficacy* and *exception/derogation*. These sorts of modifications appear particularly challenging, in that modifications in these categories make use of similar linguistic speech acts and verbs, and exhibit strong similarities in the linguistic syntactical patterns, to such an extent that to discern them is difficult for the legal expert, too. We describe the implemented system and report about an extensive experimentation on the new modificatory provisions. Results are discussed in order to improve both system's accuracy and annotation practice.

*Corresponding author.

ICAIL '13, June 10 - 14 2013, Rome, Italy

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: Text analysis; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Linguistic processing*

Keywords

Natural Language Processing, Information Extraction

1. INTRODUCTION

A wealth of efforts have been recently invested in the Artificial Intelligence and Law community to the ends of building systems for indexing, querying, searching and annotating the ever increasing amount of legal documents. Some information extraction systems analyze only texts surface elements –mostly based on numerical techniques, and representing documents through a bag-of-words approach–, e.g. identifying keywords, to compactly describe documents. However, for many tasks more sophisticated approaches are needed, such as news extraction [12], text classification [13] and summarization [16]. The aspects peculiar to the legal domain make it difficult to distinguish commonsense speech from the juridical lexicon and legal jargon, and the NLP techniques need to be applied with some more warnings and cautions [29].

Information Extraction (IE) techniques are often used to extract information useful to semantically annotate texts. In the legal domain, systems have been built that automatically identify and classify structural portions of legal documents and their intra– and inter–references [5, 22], and that produce semantic analyses [30, 31]. Various initiatives have been undertaken to render legal sources through machine understandable formats, and to devise schemas that identify legal documents elements [19]. Several tools have been developed in the last decade for assisting annotators with parsers to detect normative references and legal structure text [25, 4, 10]. However, such technologies are not yet mature enough to fully automate the annotation process,

and a good deal of human efforts are still needed. Although highly valuable, human annotation process is expensive and error-prone; this produces a strong demand for automatic tools not only to extract the structural elements of legal documents, but also to annotate them with semantic information.

Although legal language is usually known to be more regular than language *tout court*, several peculiarities make it difficult to devise IE systems for the legal domain, such as the large use of the rhetoric (e.g., metaphor and similarity), long and complex sentences, a subtle usage of common terms with meaning different from the usual one [28], the adoption of foreign terms and concepts (and the deriving issues, such as *terminological* and *conceptual misalignment*) [1], the peculiar cultural and legal tradition of each legal system (e.g. civil law *vs.* common law), the implicit norms and the legal drafting techniques with their side effects (e.g. undefined reference), the ambiguous and vague legal language that may be used as a drafting technique to guarantee the long-term persistence and flexibility of general and abstract fundamental rights [27]. Therefore, in this scenario it is not easy to apply the NLP techniques usually adopted to cope with common language.

We presently focus on a subset of all possible semantic annotations, namely the annotation of *modificatory provisions*. A modificatory provision is a change made to one or more clauses within a text (or inner partitions, such as articles, paragraphs, etc.), to the whole text along with its annexes (repeal of an entire law), or to the relations among the constituent provisions of a legal system (as when a decree-law is enacted into law). Modificatory provisions are particularly relevant, since they affect the whole normative system [14]. It should be considered, in this regard, that a poor understanding of normative modifications tends to undermine the certainty of the law, so that the changes are sometimes fragmentary and incoherent, making it difficult to clearly understand what is the law, or which one of several versions of a provision counts as law.

In past works we have explored different approaches to extract information to the ends of annotating legal documents (namely, to extract modificatory provisions): a deep-parsing approach has been carried out based on pattern matching techniques applied to parse trees [20, 18]. Also, a simple though effective scheme based on compiling regular expressions has been considered [26]. This paper has two novelty elements. To deal with the above mentioned peculiarities, and to exploit the regularities of the legal language we presently put together both approaches: we combine regular expressions with deep parsing, in the effort to overcome their respective limitations. Moreover, we substantially extend the set of modifications: while in previous work only *integration*, *substitution* and *repeal* were considered, our system presently deals with 12 further sorts of modifications. The full list includes: *annulment*, *conversion*, *derogation*, *extension*, *non-application*, *postponement*, *prorogation*, *ratification*, *implementation*, *retroactivity*, *re-enactment* and *suspension*.

The paper is structured as follows: we first provide a detailed description of temporal modifications (Section 2). We then illustrate how this knowledge has been plugged into the system to detect such modifications (Section 3); we present the experimentation devised to assess the proposed approach –involving about 12,000 files of different years, thereby ensuring adequate coverage–, discuss the results and elaborate on errors (Section 4). Conclusions will close the paper.

2. TEMPORAL MODIFICATIONS

In the following we consider legal documents encoded in XML format. In particular, we refer throughout the paper to a a standard format for Italian Legal Text, the *NormeInRete* (NIR) format.[1] The NIR format encodes the structural elements used to mark up the main partitions of legal texts, as well as their atomic parts (such as articles, paragraphs, subparagraphs, and lettered and numbered items) and any non-structured text fragment [5]. Additionally, the NIR standard includes in its Document Type Definitions a part dedicated to modifications, to implement this model in XML.

Legislative provision carries pairs of dates expressing the intervals that define its own period of force (beginning from the date of enactment, commencement date) and its period of efficacy (or come into operation) [6, 15]. We therefore find the pair (f, e), where f and e stand for a provision's force and efficacy, respectively. The periods f and e are both expressed through intervals $(start, end)$. So a provision is enriched with its temporal parameters in the markup of NIR2.2: <articolo id="art15" iniziovigore="20080301" finevigore="20100301" inizioefficacia="20080601" fineefficacia="20100301">.

Several modifications affect the two time intervals. These modifications are called *temporal modifications*. Usually these kinds of modifications do not amend the text; they rather change the range of validity of the norm respect to the two legal temporal axes: period of *force* and *efficacy*. The interval of force is the period during which a normative document (or fragment) is part of the normative system; the interval of efficacy is the period during which the provision produces its juridical effects, so that it may be applicable by the judge, and it should be respected by the citizen [15]. Sometimes the two intervals are not aligned, and this causes anomalies in the normal events flow. One of the strongest side effects of this misalignment is the *retroactivity*: that is, the case in which the efficacy period starts before the enter into force of the law.

2.1 A Taxonomy of Temporal Modifications

Previous works [23] defined a taxonomy of the legislative modifications, and also an ontology included in LKIF-core [24]. Temporal modifications can be arranged in two classes: modifications of the *force*, and modifications of the *efficacy*.

For the enter into force we have the following modifications:

- *Enter into force* (*vigenza*): usually the last article/section defines a date of enter into force, but for some special type of documents the enter into force is defined by an external legislative or regulative event (e.g., "The remaining provisions of this Act will come into force on such day as the Secretary of State may by order appoint".) or by embassy letters (e.g., international treaty) or communications (e.g. conversion of bilateral agreement). "This Regulation will enter into force on the 20th day after its publication in the Official

[1] http://www.digitpa.gov.it/standard-normeinrete

Journal".[2]

- *Postponement* (*posticipo*): the postponement of enter into force is usually applied to the fragments of the legislative document not on the all document.
- *Annulment* (*annullamento*): the Constitutional Court (for the Civil Law countries) and the High Court can deliberate decisions therein annul the legislative act or some fragment of that. In this case the act or fragment is eliminated from the legal system since the origin (*ex-tunc*). "Directive 98/43 on the approximation of the laws, regulations and administrative provisions of [...] is annulled, since those articles do not constitute an appropriate legal basis for the directive."[3]
- *Prorogation* (*proroga*): the prorogation of the enter into force occurs when the law includes itself the termination date. The legislator sometime produces a temporary act to refine or complete them.
- *Re-enactment* (*reviviscenza*): when a repealed act or fragment of act is recalled inside the legislative system without modifications; it is a renewal of the previous text. Usually this situation occurs when the cancellation resulted from an error.

For the efficacy we have the following modifications:

- *Postponement* (*posticipo*): the postponement of efficacy is usually applied to the entire document, or to simple fragments. Usually the efficacy of the act is the same date of enter into force.
- *Suspension* (*sospensione*): a suspensive provision suspends for a time interval, thus specifying both beginning and end, during which an otherwise applicable suspended target provision does not apply. Suspension of a normative provision does not affect its force. The suspension could be also *sine die*, in this case the end point is not present. Often the suspension is also expressed with a duration (e.g. 6 months). "Until 31 Dec. 2003, efficacy is suspended on Annex 4 to the Decree of the President of the Council of Ministers issued on 27 Dec. 1988, published in the Official Gazette of the Italian Republic, issue no. 4 of 5 Jan. 1989;"
- *Prorogation* (*proroga*): the prorogation of efficacy is used when a suspension of the efficacy has occurred, or if the law originally defined a termination date of the efficacy. In this case the prorogation extends the period of efficacy. "Gli effetti dell'articolo 5 del decreto-legge 7 aprile 2004, n. 97, convertito, con modificazioni, dalla legge 4 giugno 2004, n. 143, relativi all'anno 2004, sono prorogati fino al 31 dicembre 2005."
- *Retroactivity* (*retroattività*): the retroactivity modification establishes that a given normative provision becomes efficacious (operative) at some time before its entry into force.
- *Extra-activity* (*ultrattività*): this modification causes a given normative provision to remain efficacious beyond the time of its repeal. "The regional norms repealed by art. 16 continue to be applied to the procedures for distributing financial support, and such application will last until the yearly program regulated by art. 9, paragraph 2, will be approved."
- *Non-application* (*inapplicazione*): inapplication causes a normative provision to loose its efficacy (i.e., the norm can no longer be applied). It often occurs when one norm is overlapped to another one in regulating the same subject matter; in these cases, one of the two norms makes inefficacious the second one. The most common case occurred when the European Communities Law regulates a matter already regulated by National Legislation, as well as when a Regional Law introduces a detailed regulation about a topic in a National Law. Inapplication can also be linked to suspension, but it involves a more complex hierarchical arrangement among norms (e.g. principle of subsidiarity: Regional Law makes inapplicable an article of National Law and when this Regional Law will be repealed, the National law will revitalize its efficacy).

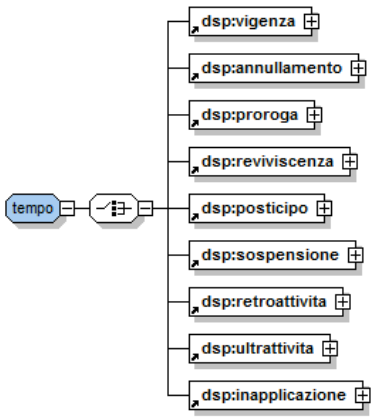

Figure 1: XSD fragment of NIR2.2 concerning the taxonomy of the temporal modifications.

2.2 Anatomy of Temporal Modifications

Following several analyses in the legal XML standardization group (CEN Metalex, NormeinRete, Akoma Ntoso), we can define the anatomy of temporal modifications as a specialized type of the more general modificatory provisions. A modificatory provision is composed by several fundamental elements that describe the action, the destination, the source of the modification, the time when to apply the modification. These elements are not marked up inside the text, but they are modeled in the metadata section called <*disposizioni*>. This partition between structural information and semantic interpretation of the provision permits multiple annotations of the same fragment without altering the original legally binding section. A temporal modificatory provision can be modeled through the following formalism.

ACTIVENORM – a provision stating a modification.

PASSIVENORM – a provision affected by the modification. The PassiveNorm can be multiple when there are many affected provisions, and it is often a problem to automatically identify all the subparts of a complex string (e.g., "Articles 3, 4, 6, paragraph 2 and 8",

[2] http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2009:008:0003:0021:EN:PDF.

[3] http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:61998J0376:EN:HTML.

where it is unclear whether the 8 identifies paragraph 8 or article 8). It is incomplete where the text includes ambiguous or not unique parameters to identify the referred provision. Sometimes the passive norm is expressed through a negative sentence (e.g., "Suspend all chapters except the first one"), making it necessary to express this through a negative proposition: Suspension (not (Chapter I)). It is not possible to have a postponement of the enter into force of the entire act, it is linguistically expressed like a postponement but it is the normal set up of the enter into force using a different rule respect to the canonical one (e.g., in Italy the enter into force starts after 15 days from the publication date).

ACTION – an action the active provision entails for the passive one. Actions are organized into the taxonomy presented in Section 2.1.

TIMES – Times refers to two intervals, the former indicating the time during which a modificatory provision is in force, and the latter the time during which it is efficacious.

DURATION – In temporal modifications it is important also to define the duration of the event, that is how long the norm is suspended or prorogated.

SPACE – A parameter used to specify a geographical area across which the modification applies (e.g., "Estonia shall be granted a temporary derogation from the application of Article 21(1)(b) and (c) until 31 December 2012").

CONDITIONS – Sometimes a norm's efficacy is conditional on an event, geographic space, or class (or domain) of application. When a modificatory provision is conditioned by an undefined event, this freezes the action until the conditional is resolved.

Often temporal modifications are located in the last part of the act (Final Provisions Section) and these clauses define the temporal parameters, among the others, of the act itself. When an ActiveNorm and a PassiveNorm are equal we have a *reflexive* modificatory provision, the provision acts on the same text with an introversion modification. This kind of modification is usually aimed at postponing a norm's application (e.g., "7. The percentage referred to in paragraph 1(d) and paragraph 3 shall from 1 January 2011 be 25%)" or at implementing an exception, condition, for restricting or expanding a norm's scope or jurisdiction. The NormeInRete (NIR) standard includes in its Document Type Definitions (DTDs) a part dedicated to modifications <*disposizioni*> that makes it possible to implement this model in XML. Below we provide an example of how a non-qualified provision can be enriched with semantic metadata (bold type) by marking it up in XML through NormeInRete.

2.3 Linguistic and Legal Ambiguities

The work focuses on the suspension, postponement (of force and efficacy time) and prorogation (of force and efficacy time) categories of temporal modification because these three sorts of provisions use similar legal linguistic expressions, and they are frequently mislead also by human experts. The goal of the work is to provide some help to the end-user fostering the XML NIR structural markup, the previous metadata information (date of publications, date

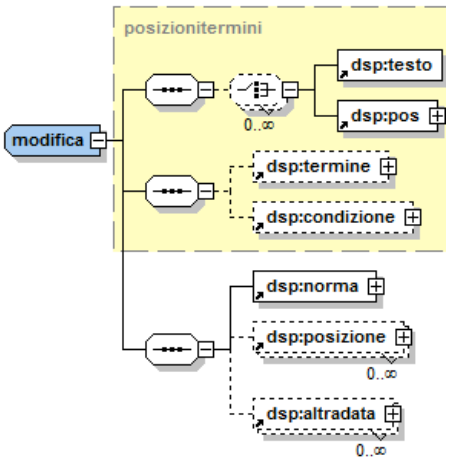

Figure 2: Modificatory provisions metadata model.

of enter into force of the act, etc.) in order to apply our analyses. The most frequent problems occur between the following categories:

1. Postponement or enter in force? The legislator is not so precise in his terminology, and often the word "in force" is used, but with the meaning of "efficacy". The "enter into force" event defines the first date of enter into force, the postponement is a second event that postpones the enter into force. One rule that we can use from the theory of law states that it is not possible to define a date of enter in force of a single article if the whole document is not set up before. So if we have an enter into force selectively for fragments, we are confident to interpret this event as postponement with respect to the original enter into force of the whole document.
2. Suspension or postponement of the efficacy? The linguistic patterns are really similar, especially for the suspension with only the final point.
3. Postponement or prorogation of efficacy? The linguistic patterns are similar; the prorogation destination should be in the status of suspended or close to be suspended. The NLP tool cannot access this information, which is stored in the XML destination files. So it is really a challenge to disambiguate these cases when the verb "apply" is used.
4. Suspension or exception? The exceptions belong to the modification of the scope and they restrict the range of cases to which a normative provision applies, as formulated in the abstract. An exception or derogation provision may be internal to a normative act or external to it. This modification may be reflexive on the same provision; that is, it restricts the scope of a rule set forth in the provision itself. Often this category of modifications conflicts with the temporal modification, especially with the suspension. "This Act shall not apply to the Faroe Islands until 2013" This provision includes typical pattern of the suspension, but the norm affects the jurisdiction (Faroe Islands) and

not the efficacy.

3. EXTRACTION OF MODIFICATIONS

In this work we substantially extend the set of modificatory provisions so far considered [18] by restructuring the overall architecture of the system: that is, instead of attempting to parse each and every sentence from the 'meaningful' XML elements, we restrict the use of the parser to some sentences. Specifically, the parser is called only in case syntactic details are essential to identify the pieces of information associated to *active* and *passive* norm (see Section 2.2), and after a filtering step based on regular expressions, with beneficial effects on both accuracy and performance.

The internal representation used by the system is in essence similar to that of [20, 18], where modifications are represented by means of semantic frames, composed by slots [11]. In this setting, retrieving a modificatory provision amounts to choosing the frame describing that modification, and to filling its slots with the correct arguments.

Before starting the main procedure to extract modifications, some preprocessing is performed. The first step of the system is the extraction of the XML nodes ⟨*corpo*⟩ and ⟨*titolo*⟩,[4] where modifications can possibly be found. In this phase a basic rewriting of the extracted text is performed to the ends of simplifying the input. For example, thousands separators are removed; typos due to wrong characters encoding such as `&#xE8` are converted into the proper characters; and NIR constants such as *rif*[5] are capitalized, so to ease their recognition by the parser as sentence subjects or objects at a later time. For the sake of simplifying the recognition through regular expressions, *Rif* constants are used to replace the corresponding text; for example, a sentence like `L'applicazione dell'<rif id="rif21" xlink:href="urn:nir:stato:decreto.legge:2003-12-23;347#art5">articolo 5 del decreto-legge n. 347 del 2003</rif> è sospesa fino al 31 dicembre 2004`[6] is replaced by the following: `L'applicazione dell'`*`Rif21`*` è sospesa fino al 31 dicembre 2004`.[7] In order to overcome the problem of dealing with coordination –which has very peculiar traits in Italian legal jargon– coordinated constant *Rif* references are collapsed, to simplify the parsing task: e.g., a sentence such as "Rif_1, Rif_2 and Rif_3 are suspended ..." is rewritten into "Rif_K are suspended ...".

Then the strings contained in the ⟨*corpo*⟩ elements are split into sentences, and each sentence is processed. All of the implemented modifications are tested through a set of regular expressions: sentences matching one or more regular expressions are marked to search the elements to fill the modification slots. A modification can result from the combination of different modifications (e.g., suspension can be described in terms of a prorogation of efficacy and of an in-application), so at this stage a sentence can be marked with multiple modification types.

We mentioned that to retrieve a modificatory provision involves filling the slots associated to the frame describing current modification. The data structure associated to the considered modification is then instantiated, and the elements *type* (that is, the type of modification) and *position* (the position of the norm describing the modification) are set to the appropriate slots.

Algorithm 1 The main control strategy implemented by the system.

```
Require: file F
for all sentences ∈ F do
  if Ratification then
    check Ratification
  end if
  if Conversion then
    check Conversion
  end if
  if Implementation then
    check Implementation
  end if
  if Extension then
    check Extension
  else
    if Suspension then
      check Suspension
    else
      if Non-Application then
        check Non-Application
      end if
      if Efficacy Prorogation then
        check Efficacy Prorogation
      else if Force Prorogation then
        check Force Prorogation
      end if
      if Postponement then
        check Postponement
      else if Retroactivity then
        check Retroactivity
      end if
    end if
  end if
  if Derogation then
    check Derogation
  end if
  if Re-enactment then
    check Re-enactment
  end if
  if Enter into Force then
    check Enter into Force
  end if
  if Annulment then
    check Annulment
  end if
end for
```

The main set of tests is provided in Algorithm 1; these

[4]The first XML element is used to denote the 'body' of the informational content of files, whilst the latter one is used to denote the 'title' of a structural partition of a norm.

[5]Standing for the Italian word 'riferimento', 'reference', a *Rif* denotes a norm. For example, a reference to the Decree Law n. 347/2003, would be encoded in the NIR format as `<rif id="rif21" xlink:href="urn:nir:stato:decreto.legge:2003-12-23;347#art5">articolo 5 del decreto-legge n. 347 del 2003</rif>`.

[6] *The efficacy of the <rif id="rif21" xlink:href="urn:nir:stato:decreto.legge:2003-12-23;347#art5">article 5 of the decree-Law n. 347/2003</rif> is suspended up to December 31st, 2004.*

[7] *The efficacy of the Rif21 is suspended up to December 31st, 2004.*

tests are used to decide about further analyses to be performed for the modification at hand. The reminder of this Section illustrates which actions are actually hidden in the `check` instructions in Algorithm 1. Legal knowledge on modification has driven the design of the main control strategy, in particular as regards as the ordering of tests, and in their nested structure that reflects the fact that linguistically close modifications need to be considered together (Algorithm 1, lines 11–31), where suspension, non-application, efficacy prorogation, postponement, retroactivity are handled. In particular, linguistic locutions describing Suspension modifications may overlap to linguistic locutions that are also used for other modifications, thereby resulting in a substantial ambiguity.

Sentences where a modification can possibly be found are parsed with the TUP parser [17]. The verbs matching the above regular expression are searched and retrieved throughout the parse tree. Parsing information is used to identify the subject and the object that indicate the norms involved in the modification. This step is particularly useful in that parsing information allows handling diathesis alternations, unveiling long-span dependencies by handling parenthetical and relative clauses. Moreover, we devised a basic solution for resorting to information provided through deictic expressions, such as for '*current* law is postponed': in these cases the identifier of current document is retrieved and used to fill the slot associated to that syntactic element.

Since salient elements and cues to identify modifications are often dispersed in long sentences, we devised a sliding windows approach. A sliding window (sized up in accord to the modification type) is defined to inspect the input sentence, based on the assumption that the meaningful elements of this modification lie within this scope: further filters are then executed, aimed at checking whether the matched elements fall within the appropriate window. The sliding window is necessary in particular for extension and derogation modifications, that are featured by more convoluted sentences, which can be hardly parsed. It is in general largely acknowledged that the difficulty inherent in parsing increases in non linear fashion in the length of the input sentence.[8] The rationale underlying the adoption of the sliding window is that if searched terms are too far from the head verb, it is more likely the parser to compute wrong dependencies. By considering only restricted fragments we focus on smaller and mostly correct fragments from the parser output. For example, while considering the derogation modification, if the matched words 'concedere' (*concede*) and 'deroga' (*derogation*) are more than 15 words apart, it is supposed that the term 'deroga' is not dependent from the verb 'concedere'. These filters ensure that some false-positive results are discarded. Furthermore, some checks can be enforced so to disambiguate between 'contiguous' modifications, such as between postponement of efficacy and modification to the extension of the norm. In particular, it may happen that the extension of the norm has been ruled out before, but some linguistic cues for this modification can be matched by filters devised for the postponement. Let us consider, e.g., the sentence: "Le disposizioni di cui al Rif1 si applicano al personale dell'Ente [...] assunto a partire dalla data [...]" (*Provisions contained in Rif1 are applied to staff members [...], recruited since date [...]*). This modification is not an extension of the norm, but it is not even a postponement; to avoid generating a false positive, it is filtered through the regular expression:

((?<!(e))si applica(no)? a(?! (decorrere|partire)))

The point is subtle, in that a normative provision may *apply to* someone/something (e.g., a category of workers or to particular circumstances), or by starting from a point in time: that is, the subcategorization frame of the Italian verb corresponding 'to apply' (*applicarsi*) allows for both an indirect object and a time span, both expressed with the same preposition.

Once groups have been identified, dates are retrieved within groups. Coherently with the pattern matched, a group can be used to identify a time interval, in which case we need to identify dates for its beginning and end; or, alternatively, the group can be used to identify a start date or an end date. Dates described with the diverse expressions used in natural language are converted into a formal representation ISO 8601 compliant, the 8-digit format *YYYYMMDD*.[9] Dates are reconstructed both based on precise expressions, such as "December 31^{st}, 2013" (in which case a date is created with value *20131231*) and on 'open' expressions such as "the first day after the publication date of this provision". In this case, based on the date of publication of the present provision, a date in the same *YYYYMMDD* format is built.

After the data formalization step has taken place, we apply further filters that build on dates. For example, such filters can be used to discriminate between the *postponement of efficacy* modification and the *retroactivity* modification: if the date of beginning of efficacy precedes the entry into force of the provision, the *postponement of efficacy* is discarded; otherwise the *retroactivity* can be discarded. To actually use dates information it is often necessary to consider contextual information, since the same linguistic locutions can be used with opposite meaning. For example, the Italian 'fino a' (*until*) can be present in the suspension modification with meaning *until* or *up to*: in this case, the associated date indicates the end of the time span during which the norm is suspended (*the norm is suspended until January 31^{th}, 2014*). Conversely, in a sentence such as *this norm has efficacy until January 31^{th}, 2014* the date is the starting point of the time interval during which the norm is suspended.

To identify the norms actually involved in the modificatory provision we then retrieve the verb in the matched pattern from the parse tree of the sentence, and we pick up the subtree rooted in that node. Interestingly enough, this needs not to be the root of the whole sentence, which in the Italian legal jargon can be rather long and complex. According to the matched pattern, we retrieve either the subject or the object, or both of them.

The final step consists of assigning the semantic frame with the information –mainly dates and *Rif*s– collected in the previous stages of the computation. All coordinate elements (previously collapsed references) that belong to the same norm are assigned to the same frame; otherwise, in case collapsed references belong to different norms, we create one

[8]For example, in contests for Italian parsers, it is customary to disregard sentences longer than 40 words [8].

[9]The standard "ISO 8601 Data elements and interchange formats – Information interchange – Representation of dates and times" was devised for the exchange of date and time-related data, `http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=40874`.

modification for each such norm.

4. EXPERIMENTATION

To assess the proposed approach we designed an extensive experimentation to analyze documents of different years, which implies dealing with slightly different information schemas (e.g., our documents refer to two different NIR DTD versions: 2.0 and 2.2), different annotation practices, and a wide range of normative sources (laws, decree-laws, further sorts of decrees, regulation statutes, etc.).

Since the system is mainly concerned with assisting legal annotators in locating and qualifying modificatory provisions, we assume that in a real setting it is preferable to provide them with the information about which sorts of modifications are present in a given document, and where to find such modifications. Further information (to fill the remaining slots of the semantic frame representing each modificatory provision) is considered less important, in that once identified a point in the document and a type of modification, by inspecting a small neighborhood the full information about the modificatory provision can be easily and quickly retrieved. Also, since we aimed at maximizing the cues provided to annotators to concentrate only on small portions of documents, we privileged the recall over the precision.

Hence, we recorded two classes of metrics: a *surface* metrics, and a *deep* one. In the surface metrics (measured through *precision* and *recall*) we look at the type of the modification and its position within the document. The deep metrics (using the *accuracy* measure) considers all elements needed to fully qualify a given modification, such as the start date and the end date of time intervals, the modified norm along with its further partitions. In the present setting, for the retrieval of modificatory provisions *MPs* we computed precision and recall as follows:

$$\text{precision} = \frac{|\{\text{relevant } MPs\} \cap \{\text{retrieved MPs}\}|}{|\{\text{retrieved MPs}\}|}$$

and

$$\text{recall} = \frac{|\{\text{relevant MPs}\} \cap \{\text{retrieved MPs}\}|}{|\{\text{relevant MPs}\}|}.$$

The deep metrics accuracy is recorded only for modifications where modification type and and its position have been correctly identified, which explains why accuracy can be higher than recall. It is computed as

$$\text{accuracy } (MP) = \frac{\#\text{ elements correctly retrieved in } MP}{\#\text{ elements in } MP}.$$

Material

We collected a dataset of files annotated by the legal experts of the University of Bologna; the whole dataset contains 12,238 files mainly dating to 2007 and 2008, as it is illustrated in Table 1. Overall 1,146 modifications are present in the dataset, distributed as depicted in Table 2.

Results and Discussion

The overall results of the experiments on all modification types listed in Table 3 are: accuracy 66%; recall 61%; precision 47%.

Given the number of modifications tested, overall the results are encouraging. To fully assess the results it is interesting to analyze the system's output, to shed light on its strengths and weaknesses. In the following we review some points that will be considered in our future work to improve the implemented system along with the annotation process.

Table 1: Files in the dataset per year.

2000	2001	2002	2003	2004	2005
1	1	0	0	2	139
2006	2007	2008	2009	2010	2011
60	5,878	5,990	119	38	10

Table 2: Number of modifications in the dataset per type.

Annulment	Conversion	Derogation
24	50	522
Extension	Non-Application	Postponement
131	23	73
Prorogation	Ratification	Implementation
77	34	60
Retroactivity	Re-enactment	Suspension
24	2	62

The XML NIR input files are rather noisy: annotation errors are frequent that prevent the system from correctly recognizing the modificatory provisions. Let us consider, e.g., the modification illustrated in Figure 3, where the XML element *Rif*, designed to annotate references (that is, further norms), is used to tag the whole sentence describing the modification (Figure 3: the *rif* with *rif id="rif67"* is used to tag a sentence corresponding to *This Agreement is valid for the period from January 1, 2004*). This way, after the rewriting step, the text of the modification –hidden in the *rif* element– can no longer be retrieved. Additionally, in the considered example, also the end term of the suspension is marked with a *rif* element (Figure 3: *rif id="rif70"*). As it can be easily seen, this excerpt does not allow to identify two modificatory provisions: a suspension and a retroactivity.

Another frequent annotation error is exemplified in the XML excerpt in Figure 4 (*Modificatory provisions settled by elements a) and b) of paragraph 28 take effect from the date July 1st, 2007*), that portraits an inconsistency between the metadata information –where two *rif* elements are present–, and the actual text annotation, where only one *rif* is marked. In this particular case, only one of the two passive norms has been annotated, thereby badly affecting system's accuracy.

Also, by inspecting the errors we noticed that the range of the linguistic expressions handled by the system can be enriched to extend its coverage of diverse locutions. For example, as regards as dates modeling, longer time spans such as years were not initially accounted for, but in the dataset considered for the experimentation we found sen-

Table 3: The accuracy results on the 4 considered modifications.

	accuracy	recall	precision
Postponement	75%	64%	51%
Prorogation	56%	72%	32%
Derogation	68%	63%	46%
Suspension	59%	62%	54%

```
<comma id="art1-com2">
  <num>2.</num>
  <corpo>
    <rif id="rif67" xlink:href="#ann1-S2601546">Il
      presente accordo ha validità per il periodo 1
      gennaio 2004</rif> -
    <rif id="rif70" xlink:href="#ann1-S2601546">31
      dicembre 2006</rif>
     Gli effetti giuridici decorrono dal giorno
    successivo alla data di stipulazione salvo diversa
    prescrizione prevista dallo stesso accordo.
  </corpo>
</comma>
```

Figure 3: Example of improper use of the tag *rif*, that makes impossible to retrieve the modificatory provision.

```
<!-- ______ METADATA ______ -->
<dsp:posticipo>
  <dsp:pos xlink:href="#art1-com29"/>
  <dsp:termine da="e2"/>
  <dsp:norma xlink:href="#ann1-S2607318">
    <dsp:sub xlink:href="urn:nir:stato:legge:
2006-12-27;296:comma.1#art1-com28-let1"/>
    <dsp:sub xlink:href="urn:nir:stato:legge:
2006-12-27;296:comma.1#art1-com28-let2"/>
  </dsp:norma>
</dsp:posticipo>


<!-- ______ ANNOTATED TEXT ______ -->
<comma id="art1-com29">
  <num>29.</num>
  <corpo> Le disposizioni introdotte dalle
    <rif id="rif116" xlink:href="#art1-com28-
let1">lettere a) e b) del comma 28</rif>
  hanno effetto a decorrere dal 1 luglio 2007. [...]
  </corpo>
</comma>
```

Figure 4: Example of inconsistency between the metadata information and the actual text annotation.

tences as "Current paragraph is applied for the three years subsequent to the entry into force of current agreement".

In several cases documents do not contain enough information to discern postponement from entry into force or from retroactivity. Let us consider, in these regards, the sentence "The norm enters into force *by starting from the actual accession of Lebanon to the World Trade Organization*" presented in Figure 5. In this case there is no explicit date to collect nor a locution useful to compute a valid date, and the information about when Lebanon's accession to WTO occurred cannot be found, thereby undermining the accuracy of the system. Of course, such cases require resorting to external knowledge; however, how to plug world knowledge into the system is out of the scope of this work.

Finally, the system also extracts non trivial modificatory provisions, such as the postponement in the sentence "*Modifications [Rif32-2, Rif32-3 and Rif32-4] will take effect after 120 days from the publication date of present deliberation*" (Figure 6). In this case, we overcome the difficulties due to coordinated references by collapsing them: since in this case all references belong to the same norm (only the article number varies), a single postponement modification is built, filled with the list of amended elements.

```
<comma id="art30-com1">
  <num>1.</num>
    <corpo> Il trattamento concesso tra le Parti per
      quanto riguarda il diritto di stabilimento e la
      prestazione di servizi si basa sui rispettivi
      impegni e sugli altri obblighi a norma
      dell'accordo generale sugli scambi di servizi
      (GATS).
      <rif id="rif51" xlink:href="#art30-com1">Questa
        disposizione</rif>
      entra in vigore a decorrere dall'adesione
      effettiva del Libano all'OMC.
    </corpo>
</comma>
```

Figure 5: Example of a case where based on the available information we cannot discern postponement from entry into force or from retroactivity.

```
<comma id="art7-com1">
  <num>1.</num>
  <corpo> Le modifiche alla normativa vigente
introdotte dalle disposizioni di cui agli
    articoli <rif id="rif32-1" xlink:href="#art2">2</
rif>, <rif id="rif32-2" xlink:href="#art3">3</rif>,
    <rif id="rif32-3" xlink:href="#art4">4</rif> e
    <rif id="rif32-4" xlink:href="#art5">5</rif>
    del presente provvedimento hanno effetto decorsi
centoventi giorni dalla data di pubblicazione della
presente delibera nella Gazzetta Ufficiale [...]
  </corpo>
</comma>
```

Figure 6: Example of a case where all modification elements are correctly extracted.

5. RELATED WORK

An early attempt to extract textual modificatory provisions from legal texts is the work described in [2, 3]. The system *Themis* was designed to simplify and speed up the legal drafting process, with particular emphasis on rewriting textual amendments, so to automatically generate the amended texts embodying the changes described in the legislative drafting. In particular, it allowed drafters to encode textual modificatory provisions in a so-called *Change Description Document*, and then to generate an amending act reflecting those changes.

The work by Bolioli and colleagues [7] focuses on automatically recognizing and encoding modificatory provisions. The authors investigated at the same time how to extract intra-document citations and how to convert the norms from a textual format into an XML format, given a specific DTD. A pilot-case is considered including about 100 modificatory provisions from a small corpus of 8 Italian laws.[10]

A project that has some commonalities with our research is SALEM [9, 32]. This system automatically annotates the modificatory provisions of NIR documents by using shallow parsing coupled to a rule-based strategy to fill the semantic frames.

A recent work on the topic of automatic consolidation has been proposed in [21]. This work builds on pattern-

[10]Also, since the Italian standard NormeInRete was being devised, the paper [7] provides the first description of a software system for the automated mark-up of Italian legal texts, funded by the AIPA (the Italian Authority for promoting the information technologies in the Italian Public Administration).

matching, and performs a basic XML preprocessing; no sort of parsing information is used. We argue that, different from the Italian case, Japanese language for describing amendments is very regular and unambiguous: indeed, the authors show that sixteen regular expressions are sufficient to capture and consolidate modificatory provisions.

6. CONCLUSIONS AND FUTURE WORK

This work proposes a system to detect modificatory provisions; it couples two techniques, deep parsing and regular expressions. The aim is to extend to temporal modifications the analysis already applied to the *textual* amendments (that is, modifications containing prescriptions about how to modify a given text). Such modifications are complex, and their recognition proved to be a stimulating task: although they are conceptually distinct, they may confuse human experts, as well, by virtue of the complexity of legal language. We have shown how similar linguistic expressions actually convey fully different meanings. We have explored how to put together the recognition of precise idioms proper of legal jargon and deep parsing, that allows for the identification of long-span dependencies, diathesis alternations, handling of parenthetical clauses.

Our system deals with nearly all kinds of modifications, while our previous efforts have been substantially limited to the class of *textual* amendments. The modifications presently considered deal with variations in the *force*, in the *efficacy*, and in the *scope* of norms. For the sake of brevity we only illustrated few cases, but the implemented system goes beyond, dealing with 12 sorts of modificatory provisions.

We have experimented all implemented modificatory provisions on a large dataset, reporting synthetic results and detailed results for postponement, prorogation, derogation and suspension. We have then analyzed errors and wrong annotations to draw some (tentative) conclusions. The system provided encouraging performances, showing that overall the proposed approach is suitable for the cases considered; but it also turned out to be in need of improvements to raise the accuracy rate.

7. REFERENCES

[1] G. Ajani, G. Boella, L. Lesmo, A. Mazzei, D. P. Radicioni, and P. Rossi. Legal Taxonomy Syllabus: Handling Multilevel Legal Ontologies. In *Proceedings of Langtech 2008*, Rome, Italy, 2008.

[2] T. Arnold-Moore. Automatically processing amendments to legislation. In *ICAIL*, pages 297–306, 1995.

[3] T. Arnold-Moore. Automatic generation of amendment legislation. In *Proceedings of the International Conference on Artificial Intelligence and Law (ICAIL)*, pages 56–62, 1997.

[4] C. Biagioli, E. Francesconi, A. Passerini, S. Montemagni, and C. Soria. Automatic semantics extraction in law documents. In *ICAIL '05: Proceedings of the 10th international conference on Artificial intelligence and law*, pages 133–140, New York, NY, USA, 2005. ACM.

[5] C. Biagioli, E. Francesconi, P. Spinosa, and M. Taddei. The NIR project: Standards and tools for legislative drafting and legal document web publication. In *Proceedings of ICAIL Workshop on e-Government:Modelling Norms and Concepts as Key Issues*, pages 69–78, 2003.

[6] N. Bobbio. *Teoria generale del diritto*. Giappichelli, Torino, 1971.

[7] A. Bolioli, L. Dini, P. Mercatali, and F. Romano. For the Automated Mark-up of Italian Legislative Texts in XML. In T. Bench-Capon, A. Daskalopulu, and R. Winkels, editors, *Legal Knowledge and Information Systems. Proceedings of Jurix 2002: The Fifteenth Annual Conference*. IOS Press, 2002.

[8] C. Bosco, S. Montemagni, A. Mazzei, V. Lombardo, F. Dell'Orletta, and A. Lenci. Evalita'09 parsing task: comparing dependency parsers and treebanks. In *Proceedings of Evalita'09*, Reggio Emilia, Italy, 2009.

[9] M. Cherubini, G. Giardiello, S. Marchi, S. Montemagni, P. Spinosa, and G. Venturi. NLP-based metadata annotation of textual amendments. In *Proceedings of Workshop on Legislative XML 2008, Jurix*, 2008.

[10] E. de Maat, R. Winkels, and T. M. van Engers. Automated Detection of Reference Structures in Law. In T. M. van Engers, editor, *Proceedings of the JURIX 2006 on Legal Knowledge and Information Systems: The Nineteenth Annual Conference* , pages 41–50, Amsterdam, 2006. IOS Press.

[11] C. J. Fillmore. Scenes-and-frames semantics. In A. Zampolli, editor, *Linguistic Structures Processing*, pages 55–79. North Holland, Amsterdam, 1977.

[12] C. J. Fillmore and C. F. Baker. Frame Semantics for Text Understanding. In *Proceedings of WordNet and Other Lexical Resources Workshop, NAACL*, Pittsburgh, June 2001.

[13] G. Forman. An extensive empirical study of feature selection metrics for text classification. *The Journal of Machine Learning Research*, 3:1289–1305, 2003.

[14] E. Francesconi and A. Passerini. Automatic classification of provisions in legislative texts. *Artificial Intelligence and Law*, 15(1):1–17, 2007.

[15] R. Guastini. *Teoria e dogmatica delle fonti*. Giuffrè, Milan, Italy, 1998.

[16] E. Hovy. Text summarization. In R. Mitkov, editor, *The Oxford Handbook of Computational Linguistics*, Oxford Handbooks in Linguistics, chapter 32, pages 583–598. Oxford University Press, Oxford, 2003.

[17] L. Lesmo. The Rule-Based Parser of the NLP Group of the University of Torino. *Intelligenza Artificiale*, 2(4):46–47, June 2007.

[18] L. Lesmo, A. Mazzei, M. Palmirani, and D. P. Radicioni. TULSI: an NLP System for Extracting Legal Modificatory Provisions. *Artificial Intelligence and Law*, 12(4):1–34, 2012.

[19] C. Lupo, F. Vitali, E. Francesconi, M. Palmirani, R. Winkels, E. de Maat, A. Boer, and P. Mascellani. General XML format(s) for legal Sources - ESTRELLA European Project. Deliverable 3.1, Faculty of Law, University of Amsterdam, Amsterdam, The Netherlands, 2007.

[20] A. Mazzei, D. Radicioni, and R. Brighi. NLP-based Extraction of Modificatory Provisions Semantics. In *Proceedings of the International Conference on Artificial Intelligence and Law, ICAIL09*, pages 50–57, Barcelona, Spain, June 2009. ACM.

[21] Y. Ogawa, S. Inagaki, and K. Toyama. Automatic consolidation of japanese statutes based on formalization of amendment sentences. In *Proceedings of the 2007 conference on New frontiers in artificial intelligence*, JSAI'07, pages 363–376, Berlin, Heidelberg, 2008. Springer-Verlag.

[22] M. Palmirani and R. Brighi. An XML Editor for Legal Information Management. In R. Traunmüller, editor, *Electronic Government*, volume 2739 of *LNCS*, pages 421–429, Berlin / Heidelberg, 2003. Springer-Verlag.

[23] M. Palmirani and R. Brighi. Time model for managing the dynamic of normative system. *Electronic Government*, pages 207–218, 2006.

[24] M. Palmirani and R. Brighi. Model regularity of legal language in active modifications. In *AICOL Workshops*, pages 54–73, 2009.

[25] M. Palmirani, R. Brighi, and M. Massini. Processing Normative References on the Basis of Natural Language Questions. In *DEXA '04 Proceedings of the Database and Expert Systems Applications, 15th International Workshop*, pages 9–12. IEEE Computer Society, 2004.

[26] L. Robaldo, L. Lesmo, and D. P. Radicioni. Compiling Regular Expressions to Extract Legal Modifications. In B. Schafer, editor, *Proceedings of the The 25th International Conference on Legal Knowledge and Information Systems*, Amsterdam, Netherlands, 2012. IOS Press.

[27] S. Rodotà. *Clausole e principi generali nell'argomentazione giurisprudenziale degli anni novanta*, chapter La tecnica legislativa per clausole generali in Italia. Cedam, Padova, 1998.

[28] R. Sacco. Lingua e diritto. *Ars Interpretandi*, 5, 2000.

[29] M. T. Sagri and D. Tiscornia. Le peculiarità del linguaggio giuridico. Problemi e prospettive nel contesto multilingue europeo. *mediAzioni*, 7, 2009.

[30] J. Saias and P. Quaresma. A Methodology to Create Legal Ontologies in a Logic Programming Based Web Information Retrieval System. *Artificial Intelligence and Law*, 12(4):397–417, 2004.

[31] C. Soria, R. Bartolini, A. Lenci, S. Montemagni, and V. Pirrelli. Automatic Extraction of Semantics in Law Documents. In C. Biagioli, E. Francesconi, and G. Sartor, editors, *Proceedings of the V Legislative XML Workshop*, pages 253–266. European Press Academic Publishing, 2007.

[32] P. Spinosa, G. Giardiello, M. Cherubini, S. Marchi, G. Venturi, and S. Montemagni. Nlp-based metadata extraction for legal text consolidation. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, ICAIL '09, pages 40–49, New York, NY, USA, 2009. ACM.